

הרצאה מספר 13

מיון לפי שדות ב-bash

האופציה `sort -k` מספר 1, מספר 2 של `sort` גורמת לכך שהמיון יתבצע לפי התחום שמוגדר על ידי זוג המספרים: ממילה מספר 1 ועד מילה מספר 2.

```
basicsys@mars~/lec6>cat F1
Jim Alchin 21 Seattle
Bill Gates 8 Seattle
Steve Jobs 246 Nevada
Scott Neally 2122 Los Angeles
```

```
basicsys@mars~/lec6>sort F1
Bill Gates 8 Seattle
Jim Alchin 21 Seattle
Scott Neally 2122 Los Angeles
Steve Jobs 246 Nevada
```

```
basicsys@mars~/lec6>sort -k 3,3 F1
Jim Alchin 21 Seattle
Scott Neally 2122 Los Angeles
Steve Jobs 246 Nevada
Bill Gates 8 Seattle
```

ניתן להוסיף את האות `n` למספר 1 או למספר 2 (או לשניהם) והמשמעות היא שהמיון לפי התחום יהיה בסדר מספרי.

```
basicsys@mars~/lec6>sort -k 3n,3 F1
Bill Gates 8 Seattle
Jim Alchin 21 Seattle
Steve Jobs 246 Nevada
Scott Neally 2122 Los Angeles
```

```
basicsys@mars~/lec6>sort -k 3,3n F1
Bill Gates 8 Seattle
Jim Alchin 21 Seattle
Steve Jobs 246 Nevada
Scott Neally 2122 Los Angeles
```

```
basicsys@mars~/lec6>sort -k 3n,3n F1
Bill Gates 8 Seattle
Jim Alchin 21 Seattle
Steve Jobs 246 Nevada
Scott Neally 2122 Los Angeles
```

```
basicsys@mars~/lec6>sort -n -k 3,3 F1
Bill Gates 8 Seattle
Jim Alchin 21 Seattle
Steve Jobs 246 Nevada
Scott Neally 2122 Los Angeles
```

ניתן להוסיף את האות `r` למספר 1 או למספר 2 (או לשניהם) והמשמעות היא שהמיון לפי התחום יהיה בסדר הפוך.

```
basicsys@mars~/lec6>sort -k 3nr,3 F1
Scott Neally 2122 Los Angeles
Steve Jobs 246 Nevada
Jim Alchin 21 Seattle
Bill Gates 8 Seattle
```

ניתן להגדיר את התחום על ידי מספר זוגות כמו בדוגמה הבאה. במקרה זה המיון מתחיל לפי התחום שמגדיר זוג המספרים הראשון, עבור השורות ששוות בתחום שמגדיר הזוג הראשון המיון ממשיך לפי התחום שמגדיר הזוג השני וכן הלאה...

```
basicsys@mars~/lec6>cat F2
Jim Alchin 21 Seattle aef
Bill Gates 21 Seattle aef
Steve Jobs 21 Nevada aaa
Scott Neally 85 Los Angeles
```

```
basicsys@m~/lec6>sort -k 3n,3n -k 4,4 -k 1,1 F2
Steve Jobs 21 Nevada aaa
Bill Gates 21 Seattle aef
Jim Alchin 21 Seattle aef
Scott Neally 85 Los Angeles
```

הפעלת סקריפט ב-bash בצורה: שם הסקריפט

כאשר מפעילים סקריפט בצורה: שם הסקריפט

נפתח תת תהליך שמריץ את הסקריפט, המשתנים של הסקריפט הם מקומיים לסקריפט ושינוי בהם לא ישפיע על המשתנים של התהליך שקרא לסקריפט.

```
basicsys@mars~/lec13>cat P1
x=9

basicsys@mars~/lec13>x=100

basicsys@mars~/lec13>P1

basicsys@mars~/lec13>echo $x
100

basicsys@mars~/lec13>y=$(P1)

basicsys@mars~/lec13>echo $x
100
```

הפעלת סקריפט ב-bash בצורה: שם הסקריפט . (נקודה)

כאשר מפעילים סקריפט בצורה: שם הסקריפט .

לא נפתח תת תהליך שמריץ את הסקריפט, אלא ששורות הסקריפט מצטרפות לתהליך הנוכחי. ולכן בצורה זו המשתנים של הסקריפט הם אותם משתנים של התהליך שקרא לסקריפט, וכל שינוי בהם ישפיע על התהליך שקרא לסקריפט.

```
basicsys@mars~/lec13> . P1

basicsys@mars~/lec13>echo $x
9
```

כתיבה לקובץ ב-awk על ידי "שם הקובץ" > print

הפקודה: "שם קובץ" > מחרוזת1 print

ב-awk פותחת את הקובץ לכתיבה (החל מתחילת הקובץ) וכותבת את מחרוזת1 בשורה הראשונה של הקובץ. אם הקובץ לא קיים הפקודה יוצרת אותו. אם הקובץ קיים אז תוכנו נדרס ולאחר ביצוע הפקודה מופיעה בו רק מחרוזת1 בשורה הראשונה.

כל פקודה נוספת מהצורה הנ"ל כותבת שורה נוספת לקובץ. ניתן לסגור את הקובץ על ידי הפקודה ("שם הקובץ") close ואז הפקודה הבאה מהסוג הנ"ל תכתוב שוב לתחילת הקובץ (ותדרוס את מה שהיה שם קודם לכן).

```
basicsys@mars~/lec13>cat F1
aaa

basicsys@mars~/lec13>cat P1
BEGIN { print "abc" >"F1"
        print "def" >"F1"
        print "gh" >"F1"
      }

basicsys@mars~/lec13>awk -f P1

basicsys@mars~/lec13>cat F1
abc
def
gh

basicsys@mars~/lec13>cat P1
BEGIN { print "abc" >"F1"
        print "def" >"F1"
        print "gh" >"F1"
        close("F1")
        print "xyz">"F1"
      }

basicsys@mars~/lec13>awk -f P1
basicsys@mars~/lec13>cat F1
xyz
```

כתיבה לקובץ ב- awk על ידי "שם הקובץ" >> print

הפקודה: "שם קובץ" >> מחרוזת1 print

ב- awk פותחת את הקובץ לכתיבה (החל מסוף הקובץ) וכותבת את מחרוזת1 לסוף הקובץ (תוכן הקובץ אינו נדרס: מחרוזת1 מצטרפת לתוכן הקובץ בסופו). אם הקובץ לא קיים הפקודה יוצרת אותו. לאחר מכן כל כתיבה מהצורה הנ"ל או מהצורה "שם קובץ" > מחרוזת1 print כותבת את המחרוזת לסוף הקובץ.

```
basicsys@mars~/lec13>cat P1
BEGIN { print "abc" >>"F1"
        print "def" >"F1"
        print "gh" >>"F1"
      }
basicsys@mars~/lec13>awk -f P1

basicsys@mars~/lec13>cat F1
xyz
abc
def
gh
```

שימוש ב-'מחרוזת2' = מחרוזת1 alias ב-bash

הפקודה: 'מחרוזת2' = מחרוזת1 alias

ב- bash גורמת לכך שבכל פעם שנרשום מחרוזת1 היא תוחלף במחרוזת2 לפני ש- bash יתחיל לבצע אותה. ניתן להשתמש בפקודה זו כדי ליצור קיצורים לפקודות נפוצות.

```
basicsys@mars~/lec13>alias m='more'

basicsys@mars~/lec13>m F1
xyz
abc
def
gh
```

bash ב-inodes

לכל קובץ ב- unix יש מבנה שנקרא inode שמכיל אינפורמציה על הקובץ שכוללת בין השאר את גודל הקובץ, ההרשאות של הקובץ, זמן יצירת הקובץ וכו'. למבנה הזה יש מצביע, שנקרא inode number שמשמש את המערכת כדי לזהות את הקובץ ולהגיע ישירות ל- inode שלו.

האופציה -i של הפקודה ls מאפשרת לראות בנוסף לשם הקובץ את ה- inode number שלו.

```
basicsys@mars~/lec13>ls -l F1
-rw----- 1 basicsys basicsys 24 Jan 14 17:51 F1
```

```
basicsys@mars~/lec13>ls -i F1
664961 F1
```

הפקודה: שם קובץ2 שם קובץ1 ln

הפקודה: שם קובץ2 שם קובץ1 ln

מוסיפה את שם קובץ2 כשם נוסף לקובץ1. לשני השמות יש אותו inode number ולכן שניהם מתייחסים לאותו קובץ. כל שינוי שנעשה בקובץ כאשר פונים אליו באחד משני השמות, מופיע באופן מידי כאשר פונים לקובץ בשמו השני. שני השמות נקראים hard links לקובץ שאליו הם מתייחסים.

```
basicsys@mars~/lec13>ln F1 F2
```

```
basicsys@mars~/lec13>ls -li F*
664961 -rw----- 2 basicsys 24 Jan 14 17:51 F1
664961 -rw----- 2 basicsys 24 Jan 14 17:51 F2
```

```
basicsys@mars~/lec13>chmod o+r F1
```

```
basicsys@mars~/lec13>ls -li F*
664961 -rw----r-- 2 basicsys 24 Jan 14 17:51 F1
664961 -rw----r-- 2 basicsys 24 Jan 14 17:51 F2
```

```
basicsys@mars~/lec13>chmod o+x F2
```

```

basicsys@mars~/lec13>ls -li F*
664961 -rw----r-x 2 basicsys 24 Jan 14 17:51 F1
664961 -rw----r-x 2 basicsys 24 Jan 14 17:51 F2

basicsys@mars~/lec13>cat F1
hello hi
aaaaaaa bbbbbb

basicsys@mars~/lec13>cat F2
hello hi
aaaaaaa bbbbbb

basicsys@mars~/lec13>echo yyyy >> F2

basicsys@mars~/lec13>cat F1
hello hi
aaaaaaa bbbbbb
YYYY

basicsys@mars~/lec13>cat F2
hello hi
aaaaaaa bbbbbb
YYYY

basicsys@mars~/lec13>rm F1

basicsys@mars~/lec13>ls -li F*
664961 -rw----r-x 1 basicsys 29 Jan 14 17:58 F2

basicsys@mars~/lec13>rm F2

basicsys@mars~/lec13>ls -li F*
ls: F*: No such file or directory

```

הפקודה: שם קובץ2 שם קובץ1 ln -s

הפקודה: מחרוזת1 שם קובץ1 ln -s

יצרת קישור לקובץ1 שמתואר על ידי מחרוזת1. הקישור הזה נקרא symbolic link ויש לו inode number שונה מזה של קובץ1. האינפורמציה שהקישור הזה מכיל היא מסלול שממנו אפשר להגיע לקובץ1. אם בעתיד נמחק את קובץ1 אז שימוש בקישור הזה שלו יגרום לשגיאה. משתמשים ב-symbolic links כדי לתת קישור לתיקיות או לקבצים שנמצאים במערכת אחרת. (אין אפשרות לתת hard links לתיקיות או לקבצים במערכת אחרת).

```

basicsys@mars~/lec13>echo "hello hi" >> F1

basicsys@mars~/lec13>echo "abcdef" >> F1

basicsys@mars~/lec13>ln F1 F2

basicsys@mars~/lec13>ln -s F1 F3

basicsys@mars~/lec13>ls -li F*
664961 -rw----- 2 basicsys 16 Jan 14 18:08 F1
664961 -rw----- 2 basicsys 16 Jan 14 18:08 F2
664965 lrwxrwxrwx 1 basicsys  2 Jan 14 18:F3->F1

basicsys@mars~/lec13>chmod o+r F3

basicsys@mars~/lec13>ls -li F*
664961 -rw----r-- 2 basicsys 16 Jan 14 18:08 F1
664961 -rw----r-- 2 basicsys 16 Jan 14 18:08 F2
664965 lrwxrwxrwx 1 basicsys  2 Jan 14 18:F3->F1

basicsys@mars~/lec13>echo yyyy >> F3

basicsys@mars~/lec13>cat F3
hello hi
abcdef
YYYY

```

```

basicsys@mars~/lec13>cat F1
hello hi
abcdef
YYYY

basicsys@mars~/lec13>cat F2
hello hi
abcdef
YYYY

basicsys@mars~/lec13>rm F1

basicsys@mars~/lec13>ls -li F*
664961 -rw----r-- 1 basicsys 21 Jan 14 18:10 F2
664965 lrwxrwxrwx 1 basicsys 2 Jan 14 18:F3->F1

basicsys@mars~/lec13>cat F3
cat: F3: No such file or directory

basicsys@mars~/lec13>rm F3

basicsys@mars~/lec13>ls -li F*
664961 -rw----r-- 1 basicsys 21 Jan 14 18:10 F2

```

בקרת תהליכים ב-bash

ניתן להפעיל תוכנית ברקע על ידי הצורה: & שם התוכנית

לדוגמה, התוכנית הבאה היא לולאה אינסופית. לאחר הפעלתה יהיה עלינו ללחוץ על `ctrl+c` כדי להפסיקה.

```

basicsys@mars~/lec13>cat P1
#!/bin/bash
i=1
while [ $i -le 2 ]
do
  y=2
done

```

נוכל להפעיל את התוכנית ברקע על ידי הפקודה הבאה:

```

basicsys@mars~/lec13>P1 &
[1] 21100

```

במצב זה נוכל להמשיך להפעיל פקודות נוספות תוך כדי כך שהתוכנית P1 ממשיכה להתבצע ברקע.

```

basicsys@mars~/lec13>ls
examples F2 lec13e.txt lec13e.txt.save P1

```

הפקודה `jobs` מאפשרת לנו לראות את התוכניות שמתבצעות כרגע.

```

basicsys@mars~/lec13>jobs
[1]+  Running                  P1 &

```

הפקודה `ps` מאפשרת לנו לראות את התהליכים שמתבצעים כרגע. ההבדל בין תהליך לבין תוכנית הוא שתוכנית אחת יכולה להפעיל כמה תהליכים. בדוגמה שלהלן אנו רואים שלושה תהליכים. אחד הוא ה-`bash`, השני הוא התוכנית P1 שרצה ברקע ומפעילה כרגע רק תהליך אחד והשלישי הוא התהליך שמבצע את הפקודה `ps`.

```

basicsys@mars~/lec13>ps
  PID TTY          TIME CMD
 20965 pts/22    00:00:00 bash
 21348 pts/22    00:00:13 P1
 21376 pts/22    00:00:00 ps

```

```

basicsys@mars~/lec13>cp P1 P2

```

```

basicsys@mars~/lec13>P1 &
[1] 21389

```

```

basicsys@mars~/lec13>P2 &
[2] 21417

```

כשיש כמה תוכניות שמתבצעות במקביל ה-`+` מציינ את התוכנית האחרונה.

```

basicsys@mars~/lec13>jobs
[1]-  Running                  P1 &
[2]+  Running                  P2 &

```

הפקודה % kill -9 מאפשרת למחוק את התוכנית האחרונה.
אפשר גם להשתמש ב- מספר תהליך kill -9 כדי למחוק תהליך מסויים.

```
basicsys@mars~/lec13>kill -9 %

basicsys@mars~/lec13>jobs
[1]-  Running                P1 &
[2]+  Killed                  P2

basicsys@mars~/lec13>jobs
[1]+  Running                P1 &

basicsys@mars~/lec13>kill -9 %

basicsys@mars~/lec13>jobs
[1]+  Killed                  P1

basicsys@mars~/lec13>cat P1

#!/bin/bash
i=1
while [ $i -le 2 ]
do
    echo abc | P2
done

basicsys@mars~/lec13>cat P2
#!/bin/bash
i=1
while [ $i -le 2 ]
do
    y=2
done

basicsys@mars~/lec13>P1 &
[1] 25997
```

```
basicsys@mars~/lec13>jobs
[1]+  Running                P1 &
```

```
basicsys@mars~/lec13>ps
      PID TTY          TIME CMD
 25660 pts/2    00:00:00 bash
 25997 pts/2    00:00:00 P1
 26026 pts/2    00:00:00 P2
 26054 pts/2    00:00:00 ps
```

```
basicsys@mars~/lec13>kill -9 26026
```

```
basicsys@mars~/lec13>./P1: line 6: 26025 Done
                26026 Killed                | P2
```

```
basicsys@mars~/lec13>ps
      PID TTY          TIME CMD
 25660 pts/2    00:00:00 bash
 25997 pts/2    00:00:00 P1
 26056 pts/2    00:00:02 P2
 26084 pts/2    00:00:00 ps
```

```
basicsys@mars~/lec13>kill -9 25997
```

```
basicsys@mars~/lec13>ps
      PID TTY          TIME CMD
 25660 pts/2    00:00:00 bash
 26056 pts/2    00:01:19 P2
 26090 pts/2    00:00:00 ps
[1]+  Killed                  P1
```